Formal Methods for Railway Disasters Prevention

Lilia Belabed, Tullio Tanzi, Sophie Coudert

▶ To cite this version:

Lilia Belabed, Tullio Tanzi, Sophie Coudert. Formal Methods for Railway Disasters Prevention. 2nd International Conference on Information Technology in Disaster Risk Reduction (ITDRR), Oct 2017, Sofia, Bulgaria. pp.161-176, 10.1007/978-3-030-18293-9_14 . hal-02280313

HAL Id: hal-02280313

https://inria.hal.science/hal-02280313

Submitted on 6 Sep 2019

Formal Methods for Railway Disasters Prevention

Lilia Belabed[1,2], Tullio Joseph Tanzi[1] and Sophie Coudert[1]

[1] Institut Mines-Telecom, Telecom ParisTech, Paris, France
[2] ENGIE-INEO UTS, Montreuil, France
{lilia.belabed, tullio.tanzi, sophie.coudert}
@telecom-paristech.fr

Abstract. Due to the increasing complexity of railway signalling systems, the design of those systems is more difficult and the demonstration of their safety can be extremely tedious. In this article, the verification and validation of railway signalling systems is investigated. We explain how railway signalling functions are designed, we show how they can be mathematically modelled using formal methods and we discuss some ways to use formal methods mechanisms to design, verify signalling systems and to prove the validity of their safety properties.

Keywords: Railway signalling systems, Formal methods, Safety verification.

1 Introduction

About two centuries ago, the railway revolutionized our lives, allowed an acceleration of exchanges and redesigned our territories. Since then, it has constantly evolved and improved, complying with two contradictory requirements: speed and safety. The railway transport typifies one of the oldest safety cultures, in which, there is the willingness to create systems and installations that are not susceptible to the risk of human error. The aim being to reduce the frequency and the consequences of accidents which can be dramatic in terms of human and financial losses, due to the speed of the trains, the number of passengers and the cost of infrastructures. Nowadays, railway accidents are very rare and their consequences are far less disastrous than they have been at the beginning of the railway era. However, the safety of railway transport is not granted. In France for example, the number of people seriously injured in a railway accident has increased by 22%, in 2016 [22]. Thus, designers must constantly adapt to technological developments to maintain a high level of safety. The railway signalling is a crucial element of this safety. Its task is to give to the driver, via well-defined codes and signals, all the needed information to safely circulate, and, via the interlocking functions, it guarantees a secured track status and inhibits the movement of track devices, such as points, while a train is traveling along a route. In addition to their safety function, signalling systems must improve operations by ensuring traffic optimization. The mission of railway signalling design offices is to offer safe and optimum solutions while meeting the economic feasibility constraints. In order to do so, they rely on experts' knowledge and experience. Nowadays design offices are facing new challenges: the increasing number of passengers that requires more efficient operations, the extension of the rail networks that leads to a congestion of the installations. Moreover, due to technological evolutions such as computerization and automation, the design of signalling systems requires the ability to combine old and new technologies.

In this context, new design and verification methodologies must be provided and those methodologies must be adapted to safety systems by offering rigor and automation, and must be adapted to the specificities of the railway field. Formal methods are suitable. Indeed, thanks to its rigorous and exhaustive nature, a formal methodology could guarantee, via proof of properties, that the designed system is consistent with the specifications.

In this paper, we want to show how formal methods could be introduced in signalling design offices to assist verification, to encourage innovation and to lighten safety demonstration processes. Section 2 describes the main issues of designing signalling principles. In Section 3 we give an overview of formal methods. In Section 4 we explain how to model an electromechanical system of signalling with formal methods through, a concrete example. In Section 5, we discuss the modelling approach and its limits.

2 Evolution, Constraints, Standards

Railway is one of the safest means of transport. In railway transport, the concept of safety is essential and it is based on four factors: regulation, staff alertness, braking devices and railway signalling. Railway regulation, expressed as standards and directives, describes all the organizational, technical and legal arrangements that govern the operations and the design processes of railway systems. The standards are regularly revised to adapt to technological changes. For instance, the standard EN50128 [1] applicable for information systems of signalling has recently been revised, in 2011.

The safety integrity level (SIL) is a quantization index of risk reduction, based on a scale of one to four, and a risk analysis defines, for each function of a system, its SIL requirement. For example, the route setting commands are SIL0 because there is no need of risk reduction, while the signal opening is SIL4 because it is a safety function. The standards IEC 61508 [2] describe the development activities and the techniques to be used to comply with the SIL level. The higher the level, the more constraining are the development activities imposed by the standards. When a system is designed, it is assigned a SIL level, which expresses a safety objective, and then, the system is evaluated by certifying bodies; compliance with applicable standards means obtaining a SIL certificate. Railway signalling, was at first rudimentary; for example, on the first railroad lines, track surveillance was carried out by humans, using signal flags, marker lights and whistles to transmit signals. Now, it is a highly precise technical field, based on modern technologies, combining electromechanical devices and computer science. Since the early days of railway, the science of accident investigation started to transform railway systems to improve their reliability and engineers introduced automation to avoid human fault. For instance, one of the first major innovations was the continuous automatic compressed-air-brake, invented in the nineteenth century and still used to date in current trains. The latter system is based on the safety principle that allows to release the brake only if it is pressurized and not damaged.

Modern systems are still designed with a view to reducing the risk of human fault. In fact, since the seventies, information technology has been introduced in operations support systems, then in interlocking systems, such as the System of solid-state interlocking (PAI). More recently, this technology takes also action to automate metro lines (e.g. System METEOR) or for predictive maintenance using Internet of Things (IoT). The emergence of all these new technologies leads to more complexity and need to be supported by modern, adapted methods.

Railway Signalling in France

The railway signalling is an information system, the function of which is to control, monitor and interlock points, s²ignals and other appliances, in order to ensure a safe train-running over track sections. The main purposes of this system are to:

- maintain a safe separating distance between trains going in the same direction,
- avoid derailment due to speed excess,
- avoid traffic in both directions on the same track (face to face),
- ensure a safe traffic at level crossings,
- prevent trains from taking conflicting routes (converging traffic lanes, traffic cut…).

Another function of railway signalling systems is to ensure optimum operations while guaranteeing safety; and the growth of the number of passengers in urban transportation networks makes it a real challenge, promoting the emergence of systems as the SACEM (Système d'Aide à la Conduite à l'Exploitation et à la Maintenance) which provides optimum speed instructions to the driver.

In France, hard-wired logic systems have been favored due to their reliability, maintainability and to the intrinsic safety of their equipment. In fact, electromechanical interlocking devices are a safe bet for the railway signalling and a good knowledge of the equipment is essential for the maintenance of a system. For this reason, computerized signalling technology has not witnessed the same success. Moreover, the implementation of computer-based systems is mostly constrained by the cost of their development because that implies to be able to prove their Safety Integrity level (SIL). In fact, meeting the requirements of standards such as EN50128 [6] in terms of resources, organization and development cycles can be difficult and expensive because it imposes, at each stage, a quantity of documents (specifications, plans…), verifications and tests, carried out by independent teams. Furthermore, the software maintainability can hardly reach the safety relay’s which is ensured by the endurance of the equipment. Nevertheless, there are some good examples of the use of digital systems such as the Computer-Controlled All-relay Interlocking (PRCI) which allows the computerized command of routes, while the interlocking and the monitoring of the routes are ensured by the safety relays NS1 [3].

Besides, the French regulation requires, for all new systems or any alteration of an existing system, to demonstrate a safety level at least equal to the safety level of the existing systems [4]. Hence, it is easier to achieve an equivalent level by using the technologies of existing systems rather than trying new technologies. Therefore, we can say that the tediousness of safety demonstration can be a slowing point to innovation which is regrettable considering that signalling systems need more innovation than they ever did before Indeed, the installations are increasingly complex, congested, making them more difficult to maintain. Maybe by optimizing logic circuits or by interfacing them with digital systems, it would be possible to reduce the quantity of equipment and, as a result, reduce wire and congestion in installations Innovation on principles of hard-wired logic can also improve the operations. A perfect example [3] is the passage of rigid transit (which allowed the setting of a route only after all the occupied transit zones of the conflicting route were released) to flexible transit (which allowed the setting of a route as soon as the convergence zone with the conflicting route was released).

The design of signalling principles is a creative task based on experts' reasoning and this reasoning is usually checked manually. In fact, the verification of these principles is a real issue for design offices because it requires specific skills and good experience and knowledge of systems and equipment. Besides, it has to be carried out by two experts with a sufficient level of independence in terms of the standard EN50126 [4], who have not been involved in the design part of the system. This whole independent organization represents a significant cost for companies. In addition, an installation cannot be tested until it is totally wired, which makes the correction of errors much more expensive as it generates much more reworking. Providing designers with modeling and verification tools that afford a theoretical support to the design choices would be a good way to reduce verification costs. As pointed by [5], the automated verification of signalling systems design, especially for the interlocking part, is an open research subject for which the challenge is to handle the growing complexity of the systems.

Formal methods are useful mathematical techniques for modelling complex system designed on a logical reasoning because they provide a verification of the consistency and the validity of this reasoning [6], through proof of properties which requires a precise statement of system’s properties. This constraint is the opportunity for the designers to unambiguously specify the essential requirements of the system. These methods offer many advantages, in addition to enhancing confidence in the safety and the efficient functioning of systems; they provide a better automation of design and verification tools. The automated proof can be done in different ways, such as model checking.

As mentioned above, digital systems have not been able to replace electromechanical interlocking. But, before considering a whole transition from so-called "classical" signalling systems to computer-based systems, we can start by modernizing the methods of verification on old technologies. Formal methods could be a way of modernization. The modeling of railway signalling systems would allow doing the verification at the same time as they are designed. Formal methods, such as B method, require this verification through the proof of properties at each refinement. Finally, there is an obvious analogy between the logic of the electromechanical signalling circuits and the Boolean logic, which makes the modeling in formal language quite feasible. This analogy will be explained in section 3.

3 Formal Methods overview

Formal methods originate in logics which is, to some extent, the science of reasoning. In ancient time, Aristote characterized well reasoning as succession of sentences respecting precise patterns. It traced the path for the reduction of reasoning to a question of shape that can be verified by machines ignoring semantics. With mathematical logic [7], languages are mathematically defined by way of formal syntax. They are provided with mathematical models, i.e. precise non-ambiguous semantics. Then proving patterns are defined on a purely syntactic base. Thus, they can be handled by computer to monitor proving activity. Mathematician proves that these patterns are sound: they only allow proving true things in semantics. It guarantees that proofs using them are sure. Computer monitoring exclude human error. Of course, provided guarantees are only valid if mathematical models are relevant with respect to real world, which can only be checked by human.

Formal methods rely on such kind of foundations. They provide a lot of logical languages and associated computerized tools. They were initially developed to support software engineering [8] and enhance software reliability. Nowadays their scope extends to many domains, kind of problems or applications. Their aim is to guarantee the behavior of systems following rigorous approaches. The choice of one method depends, of course, on how the method fits into the development process as a whole. In this paper, we do not describe or classify all the methods. We provide an overview of two approaches used in the railway domain the B method [9,10] and model checking method [11].

Both approaches consider state machines as models for mathematical semantics. States are simplified views of snapshots of real world states and state changes in models are “transitions” which can be events, actions, time... depending of approaches. When the number of state is finite, state machines (also called automata) are often graphically represented by graphs with labeled states linked by labeled arrows as transitions. For example, transition labels may express conditions constraining states changes. These models are discrete: state evolves step by step and not continuously and thus modeling of continuous systems requires discretization. Lot of applications can be modeled this way, and complementary approaches [12,13] are available for a more precise handling of continuity.

The B method enables describing machines with a language that allows comprehensive characterizations of transitions and description of properties expected from the system. And then, a support is provided to ensure and exhaustive proof of these properties. Proofs are similar to usual mathematical proof. Tools provide monitoring and assistance to human work. This proof approach can’t be fully automated but its power takes benefit of human mining. The second advantage of B method is to offer a fully guaranteed refining process: a way to move from high level models (abstract simple view of application) to low level ones (detailed view of implementation) in a rigorous way. This allows expressing and proving properties on simple and user-level models, and by refinement, to ensure that these properties hold in the final technical implementation of the system. Two variants of this approach exist. The B method is dedicated to software development and in this case, state transitions are calls of software procedures.

It has been widely used for developing certified railway software [15]. The “Event B” version considers events as transitions and its scope is more generally system modeling, and not only computer or IT domain [16].

“Model checking” denotes a family of algorithms offering automated verification for finite-state automata. The principle consists in exploring the model entirely, going through all states, to verify, through logical questions, the validity (or not) of provided expected properties. Thus, it is more proof by exhaustive inventory of cases (states) than a mathematical comprehensive proof (as proofs with B method are). The approach is mathematically sound: proved properties are sure. The advantage is automation, whereas, the limit is the size of the set of states to explore, which must be finite. Model checking is often combined with abstraction techniques (the converse of refinement), which allow to forget details in models which are not significant with respect to properties of interest. Abstraction leads to simpler models, with less state and easier to check. Moreover, computing capacity increased a lot and despite the intrinsic character of complexity, model checking approaches are nowadays relevant for many applications. In the railway domain, a lot of works [5,17-20]) studies how to apply them to the interlocking problems, which is hard to solve by a general comprehensive reasoning.

Model checking and comprehensive proof are not exclusive. For example, the second can take benefit of the first to prove some intermediate results (lemmas), and conversely. Expertise leads to choose the most efficient approaches depending on the properties to prove. A domain specific methodology may provide support to help such choices and combine results. Such a methodology may also give access to the numerous theoretical and concrete primitive and tools allowing to decompose problems and specifications in order to make proof and verification simpler following the “divide and conquer” idea. Refinement and abstraction are part of this structuring toolkit. Even though no complete methodology exists for railway, as pointed by Author’s name [8], formal methods have been applied for years in railway domain; a proof of this is the fact that European Standards CENELEC [1] applicable for development of software in railway control system requires the use of formal methods for specification, design and V&V activities for software of the highest safety and integrity level

4 Railway infrastructure modelling example

A railway signalling network is composed of different electrical equipment mainly: points, shunting signals and train detection devices. Basically, a track layout consists of, at least, two tracks and it can include many routes. A route delimits the space between two signals, it is a succession of sections to be traversed, and these sections could be points. A point is a convergence spot between two tracks; it is locked in a position allowing either to traverse a route in one track, or to traverse a junction route between two tracks. A signal can be open or closed, authorizing or banning downstream the traversing of the transit zones (route). In railway signalling, an interlocking [3] physically bans the handling of signals and equipment under any condition incompatible with the traffic safety.

The main purpose of a signalling system is to open a signal if all the conditions that allows the driver to cross it are satisfied and to close it, if, at least, one condition is missing. In order to do so, information has to be exchanged. This information is classified into two types: Supervisor's commands and local information. A supervisor is in charge of the control of the signalling system, he gives commands, for example route setting, via a user interface. On the other hand, local information gives the state of the track, for example, the position of the points, the presence of a train, etc.

In hard-wired logic systems, the signalling functions for a given network layout, are described in two complementary documents: functional diagrams and scheme plans. Both must be modeled for using a rigorous approach employing formal methods. The relationship between logic and functional diagrams is simple and direct, so we will explain it in this section. The relationship is less trivial with scheme plans and it brings some methodological questions, thus, we will only give explanations about the methodologies of the domain, in this section, and the formalization will be detailed in the following section.

4.1 Functional diagrams

Functional diagrams are relay logic circuits, i.e. electrical networks that control outputs. A function (or an output) is materialized by an electromagnetic coil and controlled by a combination of conditions represented by relays connected in series or in parallel. The set of all the functional diagrams represents the global behavior of the signalling.

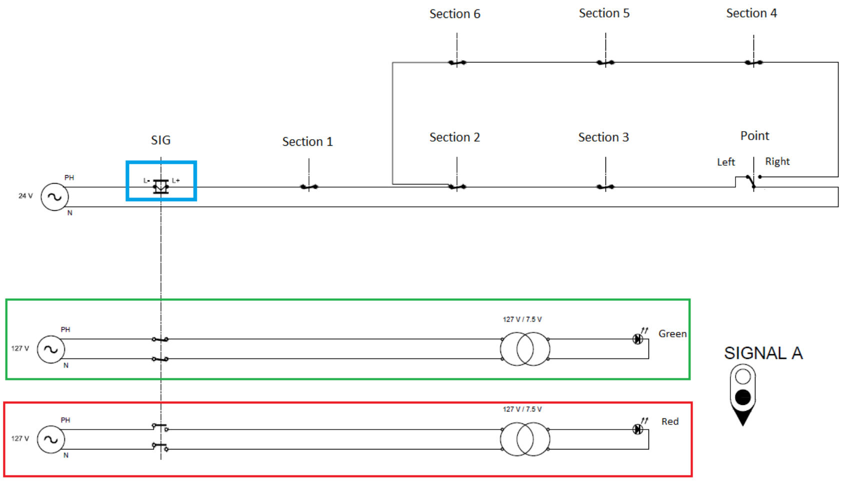

Fig. 1. Functional Diagram of Railway infrastructure example.

The functional diagram Fig. 1 describes the conditions of opening and closing a signal, respectively, allowing or disallowing the driver to cross the signal. The opening of the signal is materialized by the electromagnetic coil SIG (Surrounded by a bleu rectangle on the figure) which, once energized, closes the circuit (Surrounded by a green rectangle on the figure) that powers the green bulb of the signal and opens the circuit (Surrounded by a red rectangle on the figure) that powers the red bulb of the signal.

The coil SIG belongs to two circuits, it is energized when one of them is closed. The relays that close the circuits are the images of the state of equipment on the field. The first circuit represents one route downstream the signal A, it closes if the relays "Section 1", "Section 2", "Section 3" are closed, which means that, on the field, every section of the route downstream the signal are free (the track is clear), and if the relay "Point" is on a position that corresponds to the Left position of the point traversed by the route. The second circuit represents another route downstream the signal, it closes if the relays "Section 4", "Section 5", "Section 6" are closed and the relay "Point" is on a position that corresponds to the Right position of the point traversed by the route.

The functional diagram Fig.1 can easily be transcribed in a logical diagram, as shown in the Fig. 2 below.

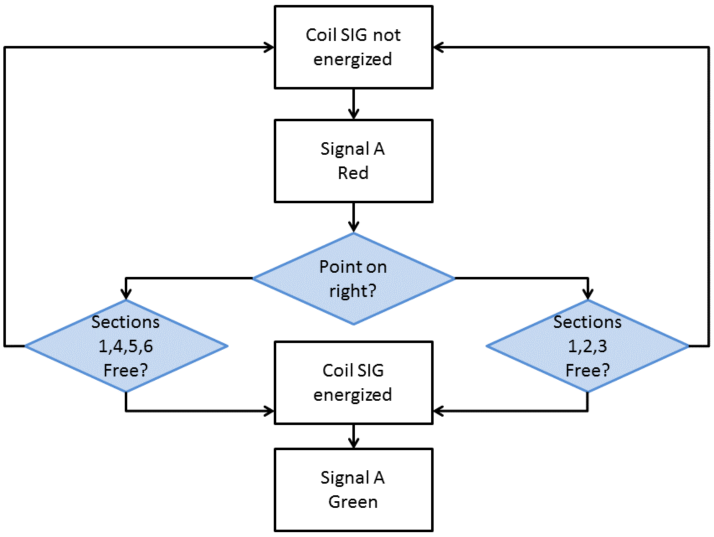

Fig. 2. Logical Diagram issue of Functional Diagram of Railway infrastructure example.

Using the logical diagram on Fig. 2, we can describe the function "Coil SIG" through Boolean logic, as showed by the following table (table 1).

With Boolean logic, le function "Coil SIG" is described by the equation below:

$$S = (A \text{ and } B \text{ and } C) \text{ or } ((\text{Not } A) \text{ and } E \text{ and } F \text{ and } G) \qquad (1)$$

This example shows how the analogy between functional diagrams and logical diagram is perfectly viable. Because they have been conceived for computer science, formal methods are suitable for signalling principles.

Table 1. "Coil SIG" description.

Function	Symbol	Boolean values
Coil SIG	S	Energized (1), Not Energized (0)
Position of the point	A	Left (1), Right (0)
Section 1	B	Free (1), Buzy (0)
Section 2	C	Free (1), Buzy (0)
Section 3	D	Free (1), Buzy (0)
Section 4	E	Free (1), Buzy (0)
Section 5	F	Free (1), Buzy (0)
Section 6	G	Free (1), Buzy (0)

4.2 Scheme plans of the example

Scheme plans comprises a track plan and various tables, among them control tables [5]. A track plan is a graphical representation of all the railway tracks in a station and control tables specify, for each route in the network layout, all the conditions for setting this route.

The figure 3 is an example of a track layout plan. It contains one point PT1 that links the track 1 and the track 2. In the case of this example, the train detection devices are axle counters. An axle counting section is marked out by at least two counting heads (CH). When a train traverses one of the counting heads which marks out a section, the number of axles of the train is recorded. This section is considered occupied until the same number of axles passes the counting head at the exit of the section. For example, the ACS1 section is marked out by the counting heads CH1 and CH3, depending on the direction of the train, each of these could be an entrance or exit point of the section. The ACS3 section is marked out by the counting heads CH3, CH5 and CH0. When the train runs the route from A to B, the counting head CH3 will be the entrance point and the CH5 will be the exit point. Whereas, for the route from D to A, the CH0 will be the entrance point and CH3 will be the exit point.

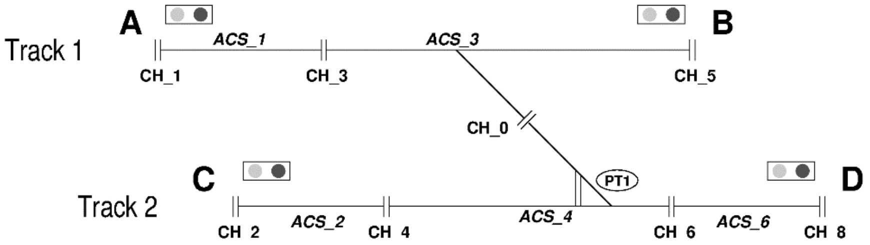

Fig. 3. Track plan corresponding to the infrastructure example.

Abbreviations of the drawings:
PT: Point
SA, SB, SC, SD: Signal A, B, C and D

CH : Counting head
ACS: Axle counting section
CPT : Command of point

The two vertical and parallel lines connecting the two switch blades of the PT1 point represent the "fouling point limit", that is to say, the limit zone where a train can stop without approaching the convergent track gauge. Furthermore, in this example, the PT1 point is a trailable and reversible point. In opposition to motorized points which receive a point's electrical command (CPT) sent by the signalling system; in a position depending on the route's direction, a trailable and a reversible point turns in a position depending on the occupied heel section. When a route traverses it in a facing mode (which is case of the routes DC and DA), the point is positioned through a manual command, by an authorized operator, at a building site respecting the safety conditions. In the case of the PT1 point:

- If the ACS2 section is occupied through the traversing of the CH2 counting head: the point turns Left.
- If the ACS3 section is occupied through the traversing of the CH3 counting head: the point turns Right.
- The default position of the point is to the Left (represented by the small line under the point).

A signalling system, for the track layout Fig. 3, must ensure the following safety features:

- Avoid collisions between trains going the same direction by prohibiting the opening of a signal if a section of the route is occupied,
- Avoid collisions between trains going in two opposite directions on the same track (face to face), by prohibiting the simultaneous opening of incompatible signals,
- Avoid collisions between trains taking conflicting routes, by prohibiting the simultaneous opening of conflicting signals.

The track table below (Table 2) inventories, for each route of the track layout, all the conditions required to open the signal upstream the route. The events that can change the state of the system are:

1. The supervisor sets a route: which can open the signal upstream the route if all the sections of the route are free and all the conflicting routes are destroyed
2. The supervisor destroys a route: which closes the signal upstream the route
3. A train traverses a counting head: which can occupy or release a section. if it occupies a section and if this section is a heel section of the point PT1, the point will turn to the corresponding position

In design offices, the verification of the two documents (functional diagrams and scheme plans) consists in checking manually and thoroughly that all the conditions described in track tables are met by the system’s behavior described in functional diagrams and also, checking that the track tables are complete according to the track layout plan. This verification could be automated using model checking, this is the topic of the following section.

Table 2. Track table.

Route's characteristics					**Conditions**		
Signal	Setted Route	Departure	Arrival	Points' position	Released sections	Destroyed incompatible routes	Destroyed conflicting routes
SA	AD	CH1	CH26	PT1 : Right	ACS1, ACS3, ACS4, ACS6	DA	BA, CD
	AB	CH1	CH15	PT1 : Left	ACS1, ACS3	BA	DA
SB	BA	CH5	CH11	PT1 : Left	ACS3, ACS1	AB	DA
SD	DC	CH6	CH22	PT1 : Left	ACS6, ACS4, ACS2	CD	AD, DA
	DA	CH6	CH11	PT1 : Right	ACS6, ACS4, ACS3, ACS1	AD	CD, BA
SC	CD	CH2	CH28	PT1 : Left	ACS2, ACS4, ACS6	DC	AD

5 Formalisation

In the scientific literature, there is many examples that confirms the suitability of model checking for the modeling of interlocking systems. Nevertheless, this method is not that easy to implement. In fact, its application can be tedious if the system is complex and it is based on the quality of the modeling which depends on human expertise. This is what we want to illustrate, in this section, by giving an overview of what can be done with model checking, and then, by justifying the importance of rigorous methodological complements.

5.1 Model checking for interlocking

To create a formal model of the system, we need to define abstract states. We use a current way to do this: we provide a finite state of state variables. A state is fully characterized by the values of these variables. Choosing relevant variables is an important aspect of modeling: they define an abstraction and they must allow to describe the system and to express the expected property with respect to this abstraction. In our pedagogical example, they must allow to represent concepts and ideas expressed in table 2, figure 3 and event description in the previous section. We choose to represent the signals, the axle count sections, a generic OUTSIDE section, the point and the routes:

- Signal_A, Signal_B, Signal_C, Signal_D accept OPEN or CLOSED as value
- Section_1, Section_2, Section_3, Section_4, Section_6, OUTSIDE accept BUSY or FREE as values
- Point accepts LEFT or RIGHT as values
- Route_AB, Route_AC, Route_BA, Route_DA, Route_CD, Route_DC accept SET or UNSET as values.

Then the temporal behavior of the system must be rigorously described in a methodical way. Various languages are usable depending on tools and formalisms. Here we use events described by two aspects: the way they modify the state and the conditions under which they can happen. Description must not only reflet reality but also provide all information required to prove the expected property, although we omit or abstract some (train direction for example), here, to simplify presentation. The three events of section X become are methodically described and something must be added to make train appear: a new event “New”.

- Set(R): set the route R. Conditions: associates sections are FREE and conflicting routes are UNSET. Modifying: Signal opening the route becomes OPEN
- Unset(R) : unset the route R. Conditions: none. Modifying: Signal opening the route becomes CLOSED
- Trav(S,S’): traverse counting head between sections S and S’. Conditions: S is BUSY and if S is OUTSIDE, then the signal associated to S’ is OPEN. Modifying: S becomes FREE and S’ becomes BUSY. If S’ is Section_3 or Section_4, Point becomes LEFT OR RIGHT, following indications provided in previous informal description.
- New: a train appears. Conditions: none. Modifying: OUTSIDE becomes BUSY.

Tools are able to build automaton from such descriptions. The single additional information they need is an initial state. They compute the set of all reachable states by successions of events, and these events are the transitions. Figure 4 provides a partial view of the example’s graph, considering an initial state without train and set route. On the figure, variables are abbreviated by their indexes. Black text is used for busy sections, set routes and open signals, and grey is used for other situations

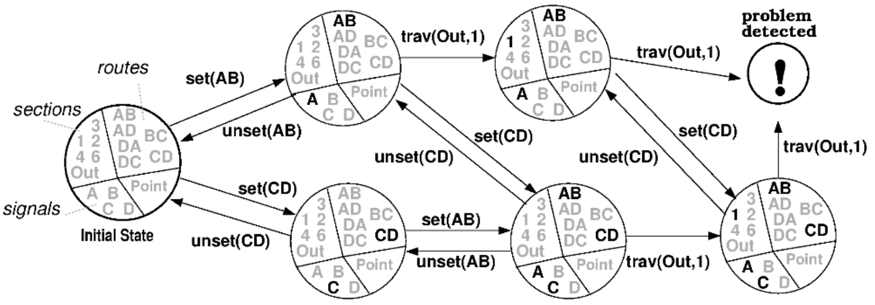

Fig. 4. Partial graph of studied system

A model checking algorithm can then automatically explore the entire automaton to verify formally expressed properties. In our case, we don’t want two trains to be on the same section at the same time, thus we could require that each time a train enters a section (except OUTSIDE), the section is free. More formally: if two states in automaton are linked by a “Trav(S,S’)” transition, then S’ is FREE in the source one, which can be obviously checked by an exhaustive exploration. If checking fails, a counterexample is generally provided to help the designer to find the error. In our example, signals remain open after a train passed them which compromise the expected property.

The suit “set(AB);trav(Out,1);trav(Out,1)” is a counterexample as shown on figure 4.

Errors can be errors in the real system. They can be also errors in the model, when for example the specifier forget some implicit information and allows then behaviors that do not exist in real world. This shows how the demand of proof helps to correct errors and construct safe solutions ([21]). The model checking avoids the risk of human error or oblivion in complex verifying. However, even if a formal verification is not susceptible to errors of reasoning, it is susceptible to errors of modeling. This is what we will explain in what follows.

5.2 Human impact

Formal approaches appeal to human expertise for various reasons. First, to avoid unnecessary complexity and to obtain optimized models, easier to implement. In fact, choosing the right variables and the right abstractions limits the number of states, in the model checking, and reduces the number of proofs. Else, the properties checked must, above all, be relevant vis-a-vis the real problems. The model must reflect the system, and the properties expressed in mathematical language must correspond to the properties that the system should ensure. For the example above, the model must reflect the signalling system’s behavior described in functional diagrams and the properties checked must be conform to the track tables. In some other areas, such as software engineering, there are design environments with graphical interfaces and various tools that help the test and visual verification of specifications. For the field of railway transport, whose experts are less accustomed to formal ratings than in computer science, such assistance is even more necessary. However, even if the methodologies used in design offices are informal, they are based on standards that provide a framework, with well-defined processes and nomenclatures, which could facilitate the formal modeling.

The specification of the signalling system, in the example above, is based on a strong hypothesis: “two trains cannot clash if they are in two different sections”. This hypothesis is true with a fixed length of trains. But, if the length grows up, the hypothesis become invalidated. Indeed, when the DA route is set (the PT1 point is previously positioned to the Right), a train (Train 1) will traverse the ACS6 section and then the ACS4 section and when the counting head CH21 counts out the last axle, the ACS4 section will be released and the ACS3 section will then be occupied. If the train stops right before passing the counting head CH21, we will be facing a problem. If the supervisor destroys the DA route and sets the CD route, another train (Train 2) could enter the point section ACS4 (left heel). Since the distance between the two tracks (track 1 and 2) is small, it is possible that the second train strikes the rear of the first train (See figure 5). This error would never have been detected by a model checker with the model defined in the previous section.

Formal methods must be used carefully and cannot replace human judgment; it shows the importance of the specification phase. In fact, to obtain a viable and exhaustive model, the system’s features should be expressed precisely and for this case of study, the property that is missing from the requirements is that we should not have the sections ACS3 and ACS4 busy at the same time, if the point PT1 is on the right position.

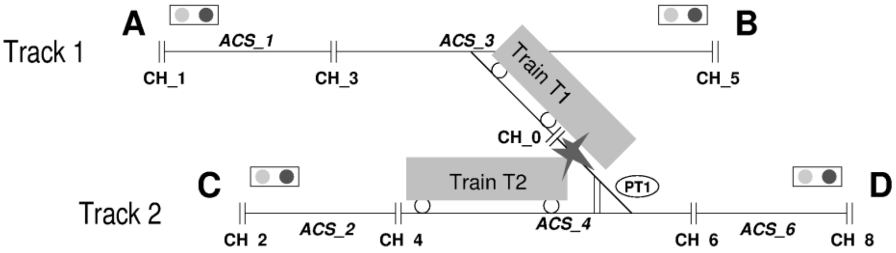

Fig. 5. Example of track plan (accident case)

The presented reasoning is based on a hypothesis: two train on different sections can't collide together. It is the base of block signal systems which has been used in railway development for a long time (long before formalization) to facilitate the design of safe interlocking. Historically, blocks exactly corresponded to physical sections. The modeling for model checking proposed above depends on this. But, as explained in section 2, railway evolves. It is more and more demanding and static block systems limits optimizations and performances. Thus, practices in the domain have also evolved from static to dynamic block systems. Now blocks do not always correspond to physical sections. They are virtual: they can comprise several sections and they can change over time. In our example, ACS3 and ACS4 should be a single block, precisely when PT1 is on the right direction. A simpler solution would have been to forbid the opening of the signal SC when the ACS3 section is occupied. This makes the routes CD and BA incompatible and we have then virtual static blocks. But this compromise is restrictive and would significantly decrease operations.

Experts of railway signaling are able to propose technical solutions to implement the dynamic model of block systems, using new equipment and technologies. As block system principles remains, the new reasoning model is quite similar to the previous one but more complex. Formal modeling may be adapted by adding state variables to characterize dynamic blocks and new events to describe the behavior of new equipment. Then their complexity increases too. Adding custom modifying to existing solutions can progressively lead to useless complexity. Thus, when reasoning paradigms evolves too much, it is required to reconsider in more depth the model, choosing new abstractions and variables in order to recover simplicity. Experience resulting from previous modelling generally makes the developing of new ones much faster, as a lot of ideas remains relevant although they are not always applied in the same way.

This example shows how it is possible to improve operations by creating new signaling principles. To verify and validate new principles, designers need reliable tools and methodologies to prove the safety of their innovative solutions. Formal methods could provide those tools and methodologies. For this, new expertise in formal methods is required. This difficulty can be overcome by simple consensus, for example the verification of the correspondence between the track tables and track layout plans can still be the task of signaling experts and the formal modeling be assigned to staff trained on the formal methods. In addition, providing intuitive graphical modeling tools could be a way for signaling experts to participate concretely to formal methods implementation.

6 Conclusion

In this paper, the application of formal methods for the design and the verification of railway signalling systems has been discussed. Considering the evolution of railway technologies and the need for increasingly efficient systems and operations, the usual means of verification are no longer appropriate. Formal methods provide solutions to deal with this context. These solutions have been detailed, as well as the reasons why a modeling a mathematical modeling of a railway system is perfectly feasible. First, an overview of formal methods has been given, focusing on two of the most widely used formal methods: B Method and Model Checking method. Next, the analogy between Boolean functions and functional diagrams has been described. Then, through an example of a track layout, the modeling process using model checking has been detailed. This example showed a way to define abstract variables to build an abstract model, in order to automate verification using algorithms of model checking. Those algorithms are not totally resistant to human errors; they are susceptible to errors of modelling. This has been illustrated by a case of accident due to equipment evolution that has not been taken into account in the model. This case allows to make a fundamental point: using formal methods does not free from the human factor. Human expertise, in the field of signaling as well as in the use of formal methods, is essential. Finally, a discussion about the way to organize a verification work by combining railway signalling expertise and formal methods knowledge, has highlighted the need for providing adapted tools dedicated to railway professions.

References

1. Standard NF EN 50128 Railway applications. Communication, signalling and processing systems. Software for railway control and protection systems.
2. Standard IEC 61508 Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems (E/E/PE, or E/E/PES).
3. Roger Rétiveau. La signalisation fériovaire. Département Edition de l'Association des Ingénieurs Anciens Elèves de l'Ecole Nationale des Ponts et Chaussées.© 1987 ISBN 2-85978-102-1.
4. Standard NF EN 50126 Railway Applications Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS).
5. Anne Elisabeth Haxthausen, Hoang Nga Nguyen, and Markus Roggenbach, Comparing Formal Verification Approaches of Interlocking.
6. S. Coudert and T. J. Tanzi, "Formal Methods for Safe Design of Autonomous Systems dedicated to risk management," (ITDRR 2016), Sofia, Bulgaria, November 16-18, 2016.
7. J.L. Krivine and G. Kreisel, Elements of mathematical logic (model theory), North Holland, Amsterdam, 1967.
8. I. Sommerville, “Chapter 27. formal methods,” in Software Engineering 9th edition, Pearson, ed., 2011.
9. J.-R. Abrial, Modeling in Event-B: System and Software Engineering. New York, NY, USA: Cambridge University Press, 1st ed., 2010.
10. J.-R. Abrial, Modeling in Event-B: System and Software Engineering. New York, NY, USA: Cambridge University Press, 1st ed., 2010.

11. E. M. Clarke, O. Grumberg, and D. A. Peled, Model checking. MIT Press, 2001.
12. J. Liu and J. Liu, “A formal framework for hybrid event b,” Electronic Notes in Theoretical Computer Science, vol. 309, pp. 3 – 12, 2014.
13. A. Platzer, “A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems,” Logical Methods in Computer Science, vol. 8, no. 4, pp. 1–44, 2012. Special issue for selected papers from CSL’10.
14. E. M. Clarke and S. Gao, “Model checking hybrid systems - (invited talk),” in Leveraging Applications of Formal Methods, Verification and Validation. Specialized Techniques and Applications - 6th International Symposium, ISoLA 2014, Imperial, Corfu, Greece, October 8-11, 2014, Proceedings, Part II, pp. 385–386, 2014.
15. sil4 railway software, http://www.clearsy.com/en/our-specific-know-how/b-method/
16. A. Fürst, Formal Development of a train control system using event-B. Theses, ETH Zurich, 2015.
17. S. Busard, Q. Cappart, C. Limbrée, C. Pecheur, and P. Schaus, “Verification of railway interlocking systems,” in ESSS 2015, Oslo, Norway, June 22, 2015., pp. 19–31, 2015.
18. L. Vu, A. E. Haxthausen, and J. Peleska, “Formal modelling and verification of interlocking systems featuring sequential release,” Science of Computer Programming, vol. 133, pp. 91–115, 2017.
19. A. E. Haxthausen and P. H. Østergaard, On the Use of Static Checking in the Verification of Interlocking Systems, pp. 266–278. Cham: Springer International Publishing, 2016.
20. M. Benerecetti, R. D. Guglielmo, U. Gentile, S. Marrone, N. Mazzocca, R. Nardone, A. Peron, L. Velardi, and V. Vittorini, “Dynamic state machines for modelling railway control systems,” Science of Computer Programming, vol. 133, Part 2, pp. 116 – 153, 2017. FTSCS 2014.
21. A. Fehnker, E. M. Clarke, S. K. Jha, and B. H. Krogh, “Refining abstractions of hybrid systems using counterexample fragments,” in Hybrid Systems: Computation and Control, 8th International Workshop, HSCC 2005, Zurich, Switzerland, March 9-11, 2005, Proceedings, pp. 242–257, 2005.
22. EPSF, http://www.securite-ferroviaire.fr/